

RECEIVED  
CENTRAL FAX CENTER  
DEC 22 2006

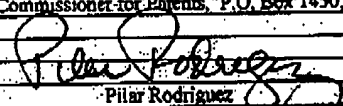
Patent

Attorney Docket No.: 2207/10119  
Assignee: Intel Corporation

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Applicant No. : 09/751,761 Confirmation No. 5065  
Applicant : Ronald D. Smith  
Filed : September 29, 2000  
For : METHOD AND APPARATUS FOR USING NEUTRAL  
INSTRUCTIONS TO PERFORM ARCHITECTURAL  
COMPARISONS  
Group Art Unit : 2183  
Examiner : David J. HUISMAN  
Customer No. : 25693

M/S: APPEAL BRIEF - PATENTS  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING	
I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office via facsimile number: (571) 273-8300 or deposited with the United States Postal Service as first class mail in an envelope addressed to:	
M/S: APPEAL BRIEF - PATENTS, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on	
Dated: December 22, 2006	 Pilar Rodriguez

**ATTENTION: Board of Patent Appeals and Interferences**

**APPEAL BRIEF**

Dear Sir:

This brief is in furtherance of the Notice of Appeal, filed in this case on September 26, 2006.

12/27/2006 HNGUYEN1 00000010 110600 09751761  
01 FC:1402 500.00 DA

Application No.: 09/751,761  
Date: December 22, 2006  
APPEAL BRIEF - PATENTS

1. **REAL PARTY IN INTEREST**

The real party in interest in this matter is Intel Corporation. (Recorded April 12, 2001; Reel/Frame 011706/0049).

2. **RELATED APPEALS AND INTERFERENCES**

There are no related appeals.

3. **STATUS OF THE CLAIMS**

Claims 20-38 are pending in the application. Claims 20, 24, 26, 33, and 37 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,643,803 (hereinafter referred to as Swoboda) in view U.S. Patent No. 5,551,050 (hereinafter referred to as Ehlig). Claims 21, 27, and 34 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Swoboda in view of Ehlig and further in view of U.S. Patent No. 5,903,768 (hereinafter Sato). Claims 22-23, 25, 28-30, 35-36, and 38 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Swoboda in view of Ehlig and further in view of U.S. Patent No. 6,285,974 (hereinafter referred to as Mandyam). Claims 31-32 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Swoboda in view of Ehlig in view of Mandyam and further in view of Hennessy and Patterson, Computer Organization and Design, 2<sup>nd</sup> Edition, 1998 (hereinafter referred to as Hennessy).

4. **STATUS OF AMENDMENTS**

The claims listed on page A-1 of the Appendix attached to this Appeal Brief reflect the present status of the claims.

Application No.: 09/751,761  
Date: December 22, 2006  
APPEAL BRIEF – PATENTS

## 5. SUMMARY OF THE CLAIMED SUBJECT MATTER

The present invention pertains to a method and apparatus for performing architectural comparisons. More particularly, the present invention pertains to neutral instruction generation that can perform error checking without changing the processor system's architectural state.

The embodiment of independent claim 20 teaches a method comprising: detecting a stall in an execution stage of a processor core (*see e.g.* page 5, lines 1-4); generating a neutral instruction (*see e.g.* page 5, lines 5-8); providing said neutral instruction to said execution stage (*see e.g.* page 5, lines 5-8); and executing said neutral instruction to ascertain an architectural state value for said processor core (*see e.g.* page 5, lines 8-9); comparing said architectural state value for said processor core to an architectural state value for a second processor core (*see e.g.* page 7, lines 12-14).

The embodiment of independent claim 26 teaches a system comprising: stall logic coupled to an execution stage of a processor core to detect a stall in said execution (*see e.g.* page 5, lines 1-4); and comparison logic coupled to said execution stage (*see e.g.* page 5, lines 11-13), wherein upon occurrence of the stall said execution stage is to execute a neutral instruction to ascertain an architectural state value for said processor core (*see e.g.* page 5, lines 8-9) and compare said architectural state value with an architectural state value for a second processor core (*see e.g.* page 7, lines 12-14).

The embodiment of independent claim 33 teaches a set of instructions residing in a storage medium, said set of instructions capable of being executed in an execution stage by a processor core for implementing a method to test the processor core, the method comprising: detecting a stall in an execution stage of a processor core (*see e.g.* page 5, lines 1-4); generating a neutral instruction (*see e.g.* page 5, lines 5-8); providing said neutral instruction to said execution

Application No.: 09/751,761  
Date: December 22, 2006  
APPEAL BRIEF - PATENTS

stage (*see e.g.* page 5, lines 5-8); executing said neutral instruction to ascertain an architectural state value for said processor core (*see e.g.* page 5, lines 8-9); and comparing said architectural state value to an architectural state value for a second processor core (*see e.g.* page 7, lines 12-14).

6. **GROUND S OF REJECTION TO BE REVIEWED ON APPEAL**

A. Are Claims 20, 24, 26, 33, and 37 rendered obvious under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,643,803 (hereinafter referred to as Swoboda) in view of U.S. Patent No. 5,551,050 (hereinafter referred to as Ehlig)?

B. Are claims 21, 27, and 34 rendered obvious under 35 U.S.C. § 103(a) as being unpatentable over Swoboda in view of Ehlig and further in view of U.S. Patent No. 5,903,768 (hereinafter Sato)?

C. Are claims 22-23, 25, 28-30, 35-36, and 38 rendered obvious under 35 U.S.C. § 103(a) as being unpatentable over Swoboda in view of Ehlig and further in view of U.S. Patent No. 6,285,974 (hereinafter referred to as Mandyam)?

D. Are claims 31-32 rendered obvious under 35 U.S.C. § 103(a) as being unpatentable over Swoboda in view of Ehlig in view of Mandyam and further in view of Hennessy and Patterson, Computer Organization and Design, 2<sup>nd</sup> Edition, 1998 (hereinafter referred to as Hennessy)?

Application No.: 09/751,761  
Date: December 22, 2006  
APPEAL BRIEF – PATENTS

## **7. ARGUMENT**

**A. Claims 20, 24, 26, 33, and 37 are improperly rejected under 35 U.S.C. §103(a) as being unpatentable over Swoboda in view Ehlig.**

**i. Swoboda does not teach several elements of appellant's claims.**

Page 3 section 6b of the Office Action dated June 26, 2006 asserts that Swoboda teaches a neutral instruction as claimed by appellant, but a careful reading of the Swoboda reference shows that the instruction referenced in the Office Action is not in fact a neutral instruction. The Abstract of Swoboda contains the following description:

While running or suspended, the emulation circuitry can jam an instruction into the instruction register of the processor to cause processor resources to be read or written on behalf of the emulation circuitry.

Appellant's independent claim 20 contains the limitation of "providing said neutral instruction to said execution stage." The antecedent basis for "said execution stage" is "an execution stage of a processor core." Therefore, it is clear from appellant's claim that the neutral instruction is being executed by the processor core. Examiner asserts that the reading of a processor resource referenced in Swoboda teaches the "executing said neutral instruction . . .," but the read instruction in Swoboda is not being executed by the processor. Instead, unlike in appellant's claim, the instruction is being executed by the emulation circuitry, which is separate from the processor.

Additionally, appellant asserts that Swoboda does not teach "executing said neutral instruction to ascertain an architectural state value for said processor core." On page 3 at section 6d, the Office Action asserts that claim 1 and the abstract of Swoboda teach this element, but appellant cannot find anywhere in the cited sections of Swoboda where ascertaining an architectural state value is taught. The office action states that the cited sections of Swoboda

Application No.: 09/751,761  
Date: December 22, 2006  
APPEAL BRIEF – PATENTS

teach receiving “a value,” but the Office Action does not show what the value is. It appears Swoboda is referring to receiving instructions, not architectural state values of processor cores. The Office Action also does not show that this undefined “value” is being received as the result of executing a neutral instruction.

For at least the reasons above, appellant asserts that the Swoboda reference does not teach what Examiner claims it teaches, and as a result, the rejection of independent claim 20 under 35 U.S.C. § 103(a) should be reversed. Independent claims 26 and 33 contain substantially similar limitations, and therefore, the rejection of claims 26 and 33 under 35 U.S.C. § 103(a) should likewise be reversed. Dependent claims 24 and 37 depend from independent claims 20 and 33, and therefore, their rejection under 35 U.S.C. § 103(a) should also be reversed.

**ii. There exists no motivation to combine Swoboda and Ehlig.**

Page 3 section 6e of the Office Action dated June 26, 2006 admits that Swoboda does not teach comparing the architectural state value for a processor core to an architectural state value for a second processor core. The Office Action claims this deficiency is made up for in Ehlig, which according to the Office Action, teaches “the concept of comparing first processor data obtained from emulation circuitry to second processor data obtained from emulation circuitry.” The Examiner, however, has not shown that the processor data in Ehlig represents an architectural state value of a processor core.

Even assuming, *arguendo*, that the processor data described in Ehlig teaches what Examiner states it does, there exists no motivation to combine Swoboda and Ehlig. Swoboda teaches a processor that can enter a debug suspend state in response to a debug event and then leave the debug suspend state in order to resume normal operation. *See* Swoboda at column 2, lines 53-60. Swoboda targets the process of debug where debug is defined as being able to

Application No.: 09/751,761  
Date: December 22, 2006  
APPEAL BRIEF – PATENTS

detect the effects of and alter the results of the execution of software on the processor. *See* Abstract. The debugging circuitry taught by Swoboda is user controlled. *See e.g.* Swoboda, column 4, lines 52-64.

Ehlig teaches synchronizing redundant processors so that data output to memory for the multiple processors can be compared, thus increasing the fault tolerance by having the processors vote. The examiner states that it would have been obvious to one of ordinary skill in the art to add a redundant processor to the disclosure of Swoboda, but the abstract of Swoboda states that the invention described is optimized for devices such as wireless telephones. Size, power consumption, cost, and many other factors make adding redundant processors to the invention described in Swoboda impractical.

Additionally, the manual debugging method taught by Swoboda is not readily compatible with a redundant processor system like that in Ehlig. The debugging method in Swoboda allows for a user to manually “fix” problems that processor’s encounter. The processors in Ehlig are all synchronous. If one processor needs to enter a debug state, as described in Swoboda, then all the other processors would be forced to do the same. Forcing all the processors to stop because of an error on one processor would defeat one of the primary purposes of having a redundant processor system in the first place.

Because the Examiner has failed to make a prima facie case that one of ordinary skill in the art would be motivated to combine Swoboda and Ehlig and because the references in fact teach away from combining, the rejection of claims 20, 24, 26, 33, and 37 under 35 U.S.C. § 103(a) should be reversed.

Application No.: 09/751,761  
Date: December 22, 2006  
APPEAL BRIEF - PATENTS

**B. Claims 21, 27, and 34 are improperly rejected under 35 U.S.C. § 103(a) as being unpatentable over Swoboda in view of Ehlig and further in view of Sato.**

Appellant asserts that dependent claims 21, 27, and 34 are allowable as depending from allowable independent claims 20, 26, and 33. Accordingly, appellants request the rejection under 35 U.S.C. § 103(a) be reversed.

**C. Claims 22-23, 25, 28-30, 35-36, and 38 are improperly rejected under 35 U.S.C. § 103(a) as being unpatentable over Swoboda in view of Ehlig and further in view of Mandyam.**

Appellant asserts that dependent claims 22, 23, 25, 28-30, 35-36, and 38 are allowable as depending from allowable independent claims 20, 26, and 33. Accordingly, appellants request the rejection under 35 U.S.C. § 103(a) be reversed.

**D. Claims 31-32 are improperly rejected under 35 U.S.C. § 103(a) as being unpatentable over Swoboda in view of Ehlig in view of Mandyam and further in view of Hennessy.**

Appellant asserts that dependent claims 31-32 are allowable as depending from allowable independent claim 26. Accordingly, appellants request the rejection under 35 U.S.C. § 103(a) be reversed.

### **CONCLUSION**

In view of the arguments above, appellant assert that independent claim 20, 26, and 31 are in condition for allowance. Appellant further asserts that dependent claims 21-25, 27-32 and 34-38 are allowable as depending from allowable independent claims. Appellant therefore respectfully requests that the Board of Patent Appeals and Interferences reverse the Examiner's decision rejecting claims 20-38, and direct the Examiner to pass the case to issue.



Application No.: 09/751,761  
Date: December 22, 2006  
APPEAL BRIEF - PATENTS

**RECEIVED**  
**CENTRAL FAX CENTER**  
**DEC 22 2006**

The Examiner is hereby authorized to charge the appeal brief fee of \$500.00 and any additional fees which may be necessary for consideration of this paper to Kenyon & Kenyon Deposit Account No. 11-0600.

Respectfully submitted,

KENYON & KENYON LLP

Date: December 22, 2006

By: 

Jeffrey R. Joseph  
(Reg. No. 54,204)  
Attorneys for Intel Corporation

KENYON & KENYON LLP  
333 West San Carlos St., Suite 600  
San Jose, CA 95110

Telephone: (408) 975-7500  
Facsimile: (408) 975-7501

**APPENDIX**

(Brief of Appellant Ronald D. Smith  
U.S. Patent Application Serial No. 09/751,761)

**8. CLAIMS ON APPEAL**

1-19 (Cancelled)

20. A method comprising:

detecting a stall in an execution stage of a processor core;

generating a neutral instruction;

providing said neutral instruction to said execution stage; and

executing said neutral instruction to ascertain an architectural state value for said

processor core;

comparing said architectural state value for said processor core to an architectural state value for a second processor core.

21. The method of claim 20 wherein said neutral instruction is generated when a plurality of instructions are generated by a compiler.

22. The method of claim 20 wherein said neutral instruction is generated by a No-operation (NOP) pseudo-random generator.

23. The method of claim 22 wherein the execution of said neutral instruction causes said processor core to access a value stored in a register in said processor core.

Application No.: 09/751,761

Response to Final Office Action dated: September 26, 2006

Final Office Action dated: June 26, 2006

24. The method of claim 20 wherein the execution of said neutral instruction causes said processor core to access a value stored in a register in said processor core.
25. The method of claim 20 wherein said neutral instruction is generated by a post-processor device.
26. A system comprising:  
stall logic coupled to an execution stage of a processor core to detect a stall in said execution; and  
comparison logic coupled to said execution stage, wherein upon occurrence of the stall said execution stage is to execute a neutral instruction to ascertain an architectural state value for said processor core and compare said architectural state value with an architectural state value for a second processor core.
27. The system of claim 26 wherein said neutral instruction is generated by a compiler.
28. The system of claim 26 wherein:  
said neutral instruction is generated by a No operation (NOP) pseudo-random generator.
29. The system of claim 28 wherein the processor core includes a register and the execution of said neutral instruction causes said processor core to access a value stored in the register in said processor core.

Application No.: 09/751,761

Response to Final Office Action dated: September 26, 2006

Final Office Action dated: June 26, 2006

30. The system of claim 29 wherein said neutral instruction includes ORing the contents of said register with itself.

31. The system of claim 29 wherein said neutral instruction includes ANDing the contents of said register with all binary 1 values.

32. The system of claim 29 wherein said neutral instruction includes ORing the contents of said register with all binary 0 values.

33. A set of instructions residing in a storage medium, said set of instructions capable of being executed in an execution stage by a processor core for implementing a method to test the processor core, the method comprising:

detecting a stall in an execution stage of a processor core;

generating a neutral instruction;

providing said neutral instruction to said execution stage;

executing said neutral instruction to ascertain an architectural state value for said processor core; and

comparing said architectural state value to an architectural state value for a second processor core.

34. The set of instructions of claim 33 wherein in said method said neutral instruction is generated when a plurality of instructions are generated by a compiler.

Application No.: 09/751,761

Response to Final Office Action dated: September 26, 2006

Final Office Action dated: June 26, 2006

35. The set of instructions of claim 33 wherein in said method said neutral instruction is generated by a No-operation (NOP) pseudo-random generator.

36. The set of instructions of claim 35 wherein in said method the execution of said neutral instruction causes said processor core to access a value stored in a register in said processor core.

37. The set of instructions of claim 33 wherein in said method the execution of said neutral instruction causes said processor core to access a value stored in a register in said processor core.

38. The set of instructions of claim 33 wherein in said method said neutral instruction is generated by a post-processor device.

Application No.: 09/751,761

Response to Final Office Action dated: September 26, 2006

Final Office Action dated: June 26, 2006

**9. EVIDENCE APPENDIX**

No further evidence has been submitted with this Appeal Brief.

- A14 -

95500.1

Application No.: 09/751,761

Response to Final Office Action dated: September 26, 2006

Final Office Action dated: June 26, 2006

**10. RELATED PROCEEDINGS APPENDIX**

Per Section 2 above, there are no related proceedings to the present Appeal.

- A15 -

95500.1